

1 Amendment "C"

2 Amendments to the Claims:

3 Please amend claims 1, 3, 9, 17, 21, 29, 31 and 37 as indicated below. The
4 state of the claims following this Amendment "C" is as follows:

5
6 Claim 1 (currently amended). A method for automated testing of a graphical user
7 interface (GUI) of a program, said method comprising:

8 creating a test case file comprising a plurality of test steps in a text
9 format wherein the test steps are not written in an interpreted computer programming
10 language and wherein each test step comprises an object, an action, and an
identification reference; and

11 executing a test harness with said test case file as input to said test
12 harness, said test harness configured to execute one of a plurality of automated
13 tests in response to one of a plurality of test steps, each automated test configured
14 to test a corresponding user interface element of said program through a GUI map,
15 said GUI map configured to define a logical name for each user interface element of
said program.

16 Claim 2 (cancelled.)
17

18 Claim 3 (currently amended). The method for automated testing of a GUI of a
19 program according to claim 1, wherein each test step further comprises an optional
20 field value.

21 Claim 4 (original). The method for automated testing of a GUI of a program
22 according to claim 3, wherein each test step further comprises an error recovery
23 value.
24
25

1 Claim 5 (original). The method for automated testing of a GUI of a program
2 according to claim 1, further comprising:

3 generating said GUI map of said program by extracting a logical name,
4 a physical name, an identification, and an ordinal value for each user interface
5 element of said program.

6 Claim 6 (original). The method for automated testing of a GUI of a program
7 according to claim 1, further comprising:

8 generating said GUI map of said program from one of a prototype of
9 said program, a design document of said program and an earlier version of said
10 program.

11 Claim 7 (original). The method for automated testing of a GUI of a program
12 according to claim 1, wherein:

13 each automated test is further configured to retrieve and to execute at
14 least one of a plurality of associated reusable functions in response to said one of
15 said plurality of test steps.

16 Claim 8 (original). The method for automated testing of a GUI of a program
17 according to claim 1, further comprising:

18 outputting results of the execution of said plurality of automated tests in
19 response to said test file.

20 (Continued on next page.)
21
22
23
24
25

1 Claim 9 (currently amended). A system for automated testing of a graphical user
2 interface (GUI) of an application, said system comprising:

3 at least one processor;

4 a memory coupled to said at least one processor;

5 a test harness residing in said memory and executed by said at least
6 one processor, wherein said test harness is configured to execute one of a plurality
7 of automated tests in response to one of a plurality of test steps of a text format test
8 data file wherein the test steps of the text format test data file are not written in an
9 interpreted computer programming language and wherein each test step comprises
10 an object, an action, and an identification reference, each automated test configured
11 to test a corresponding user interface element of said application through a GUI
12 map, said GUI map configured to define a logical name for each user interface
13 element of said application.

14 Claim 10 (cancelled.)

15 Claim 11 (currently amended). The system for automated testing of a GUI of an
16 application according to claim 9, wherein each test step further comprises an
17 optional field value.

18 Claim 12 (original). The system for automated testing of a GUI of an application
19 according to claim 11, wherein each test step further comprises an error recovery
20 value.

21 Claim 13 (original). The system for automated testing of a GUI of an application
22 according to claim 9, wherein said GUI map of said application is generated with a
23 GUI analyzer configured to extract a logical name, a physical name, an identification
24 and an ordinal value for each user interface element of said application.

25 Claim 14 (original). The system for automated testing of a GUI of an application
according to claim 9, wherein said GUI map of said application is generated from
one of a prototype of said application, a design document of said application, and an
earlier version of said application.

1 Claim 15 (original). The system for automated testing of a GUI of an application
2 according to claim 9, wherein each automated test is further configured to retrieve
3 and to execute at least one of a plurality of associated reusable functions in
4 response to said one of said plurality of test steps.

5 Claim 16 (original). The system for automated testing of a GUI of an application
6 according to claim 9, wherein said test harness is further configured to generate an
7 output file configured to contain results of said execution of said plurality of
8 automated tests in response to said test file.

9 Claim 17 (currently amended). A computer readable storage medium on which is
10 embedded one or more computer programs, said one or more computer programs
11 implementing a method for automated testing of a graphical user interface (GUI) of
12 an application, said one or more computer programs comprising a set of instructions
13 for:

14 creating a test case file comprising a plurality of test steps in a text
15 format wherein the test steps are not written in an interpreted computer programming
16 language and wherein each test step comprises an object, an action, and an
identification reference; and

17 executing a test harness with said test case file as input to said test
18 harness, said test harness configured to execute one of a plurality of automated
19 tests in response to one of a plurality of test steps, each automated test configured
20 to test a corresponding user interface element of said program through a GUI map,
21 said GUI map configured to define a logical name for each user interface element of
said program.

22 Claim 18 (original). The computer readable storage medium in according to
23 claim 17, said one or more computer programs further comprising a set of
24 instructions for:

25 generating said GUI map of said program by extracting a logical name,
a physical name, an identification, and an ordinal value for each physical element of
said program.

1 Claim 19 (original). The computer readable storage medium in according to
2 claim 17, said one or more computer programs further comprising a set of
3 instructions for:

4 outputting an output file configured to contain results of the execution
5 of said plurality of automated tests in response to said test file.

6 Claim 20 (original). The computer readable storage medium in according to
7 claim 17, wherein said one or more computer programs further comprising a set of
8 instructions for:

9 each automated test further configured to retrieve and to execute at
10 least one of a plurality of associated reusable functions in response to said one of
11 said plurality of test steps.

12 Claim 21 (currently amended). A method for automated testing of a graphical user
13 interface of a program, comprising:

14 receiving a test case file to a test harness wherein the test case file is in a text
15 format and wherein the test case file is not written in an interpreted programming
16 language;

17 reading the test case file;

18 determining a test step based on the test case file wherein each line of the
19 test case file comprises an object, an action, and an identification reference which is
20 interpreted by the software harness as a step of a test of the graphical user interface
21 of the program; and

22 invoking an automated test routine from the test harness responsive to the
23 test step determination wherein each automated test routine is configured to test a
24 corresponding physical user interface element through a GUI map of the program.

25 Claim 22 (cancelled.)

Claim 23 (previously presented). The method of claim 21, wherein each test step
further comprises a field value.

1 Claim 24 (previously presented). The method of claim 23, wherein each test step
2 further comprises an error recovery value.

3 Claim 25 (previously presented). The method of claim 21, further comprising:
4 generating the GUI map of the program by extracting a logical name, a
5 physical name, an identification, and an ordinal value for each user interface element
6 of the program.

7 Claim 26 (previously presented). The method of claim 21, further comprising:
8 generating the GUI map of the program from one of a prototype of the
9 program, a design document of the program and an earlier version of the program.

10 Claim 27 (previously presented). The method of claim 21, wherein each automated
11 test routine is further configured to retrieve and to execute at least one of a plurality
12 of associated reusable functions in response to the test step.

13 Claim 28 (previously presented). The method of claim 21, further comprising:
14 generating an output file of the results of the execution of the
15 automated test.

16 Claim 29 (currently amended). A system for automated testing of a graphical user
17 interface (GUI) of a program, comprising:

18 at least one processing means;
19 computer readable memory means which is readable by the
20 processing means, the computer readable memory means containing a test harness
21 program comprising a series of computer executable steps configured to cause the
22 processing means to:

23 control the reception of a test case file to a test harness wherein
24 the test case file is in a text format and wherein the test case file is not
25 written in an interpreted programming;
read the test case file;
determine a test step based on the test case file wherein each
line of the test case file comprises an object, an action, and an

1 identification reference which is interpreted by the software harness as
2 a step of a test of the graphical user interface of the program; and
3 invoke an automated test routine from the test harness
4 responsive to the test step determination wherein each automated test
5 routine is configured to test a corresponding physical user interface
6 element through a GUI map of the program.

7 Claim 30 (cancelled.)

8 Claim 31 (currently amended). The system of claim 30 29, wherein each test step
9 further comprises a field value.

10 Claim 32 (previously presented). The system of claim 31, wherein each test step
11 further comprises an error recovery value.

12 Claim 33 (previously presented). The system of claim 29, wherein the GUI map of
13 the program is generated with a GUI analyzer configured to extract a logical name, a
14 physical name, an identification and an ordinal value for each user interface element
15 of the program.

16 Claim 34 (previously presented). The of claim 29, wherein the GUI map of the
17 program is generated from one of a prototype of the program, a design document of
18 the program, and an earlier version of the program.

19 Claim 35 (previously presented). The system of claim 29, wherein each automated
20 test routine is further configured to retrieve and to execute at least one of a plurality
21 of associated reusable functions in response to the test step.

22 Claim 36 (previously presented). The system of claim 29, wherein the test harness
23 is further configured to generate an output file of the results of the execution of the
24 test case file.
25

1 Claim 37 (currently amended). A computer readable medium storing a computer
2 program to implement automated testing of a graphical user interface (GUI) of a
3 program, comprising:

4 computer readable code to control the reception of a test case file to a
5 test harness wherein the test case file is in a text format and wherein the test
6 case file is not written in an interpreted programming;

7 computer readable code to read the test case file;

8 computer readable code to determine a test step based on the test
9 case file wherein each line of the test case file comprises an object, an action,
10 and an identification reference which is interpreted by the software harness as
11 a step of a test of the graphical user interface of the program; and

12 computer readable code to invoke an automated test routine from the
13 test harness responsive to the test step determination wherein each
14 automated test routine is configured to test a corresponding physical user
15 interface element through a GUI map of the program.

16 Claim 38 (previously presented). The computer readable medium of claim 37,
17 further comprising:

18 computer readable code to generate said GUI map of said program by
19 extracting a logical name, a physical name, an identification, and an ordinal
20 value for each physical element of said program.

21 Claim 39(previously presented). The computer readable medium of claim 37, further
22 comprising:

23 computer readable code to generate an output file of the results of the
24 execution of the test case file.

25 Claim 40 (previously presented). The computer readable code of claim 37, wherein
each automated test routine includes computer readable code further configured to
retrieve and to execute at least one of a plurality of associated reusable functions in
response to said one of said plurality of test steps.

(End of Amendment "C".)